CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF TECHNOLOGY AND ENGINEERING (FTE)

Chandubhai S. Patel Institute of Technology (CSPIT) &

Devang Patel Institute of Advance Technology and Research (DEPSTAR)

ACADEMIC YEAR: 2025-26

Practical List

Subject: Social Network Analysis (OCCSE4002)

<table>
<tr><td>Practical Number</td><td></td><td>CO/PO</td></tr>
<tr><td>1</td><td>Problem Definition
Extract and build large real-world social networks from public Twitter APIs.

Tasks:
Build a directed weighted graph using hashtags or mentions.
Store and pre-process data using a NoSQL database (e.g., MongoDB).
Visualize stats like in-degree, out-degree, and density.
Graph should contain at least 1000 nodes.
Key Questions / Analysis / Interpretation to be evaluated during/after Implementation
How is data extracted using APIs like Tweepy or Pushshift?
How is the graph structure created and what properties are measured?
How is MongoDB used for storing and pre-processing social media data?
Supplementary Problems (For Fast Learners)
Extract and analyse user interaction networks over time.
Compare different types of graphs (mention vs hashtag based).
Key Skills to be addressed –
API data extraction
NoSQL database handling
Network modelling and visualization
Data pre-processing</td><td>CO1/PO1</td></tr>
</table>

	Applications – ● Social media influence analysis ● Community detection in online discussions ● Trend and sentiment propagation **Learning Outcome –** Students will learn to work with real-world social data, build and visualize network graphs, and apply network analysis metrics.	
	Dataset/Test Data (Source and Description If Applicable) - Data extracted using Tweepy (Twitter API) or Pushshift (Reddit API).	
	Tools/Technology To Be Used - Python, Tweepy / Pushshift API, NetworkX, MongoDB	
	Total Hours of Problem Definition Implementation – 4 Hours **Total Hours of Engagement** = Implementation + modification + testing: approx. 7-8 hours	
	Post Laboratory Work Description – ● Prepare a report including data source, graph creation steps, visualization, and network statistics. ● Submit Python scripts and screenshots of graph output.	
	Evaluation Strategy Including Viva ● Viva based on API usage and graph structure ● Evaluation of report and output ● Accuracy and size of graph ● Use of MongoDB for data handling	
2	**Problem Definition:** Extract tweets and construct user interaction networks (mentions or retweets). **Tasks:** Authenticate API, fetch tweets, build graph. **Key Questions / Analysis / Interpretation to be evaluated during/after Implementation:** How are edges formed? What challenges in real-time data extraction? **Supplementary Problems (For Fast Learners):** Perform hashtag frequency analysis.	CO1/PO1

	Key Skills to be addressed: API interaction, JSON parsing, graph formation. **Applications:** Social media analysis, hashtag tracking. **Learning Outcome:** Extract and construct networks from live Twitter data. **Dataset/Test Data (Source and Description If Applicable):** Twitter API **Tools/Technology To Be Used:** Python, Tweepy, NetworkX **Total Hours of Problem Definition Implementation:** 4–5 hours **Total Hours of Engagement = Implementation + modification + testing:** approx. 7–8 hours **Post Laboratory Work Description:** Submit data file, graph visualizations, and scripts. **Evaluation Strategy Including Viva:** Viva on API usage and data pipeline.	
3	**Problem Definition** Multi-Centrality Analysis on Large Networks. Implement and compare centrality metrics on scale-free and random networks. **Tasks:** ● Apply multiple centrality algorithms: PageRank, Katz, Eigenvector, Closeness, and Betweeness. ● Use libraries such as NetworkX, SNAP, 2and igraph. ● Compare time complexities and analyze scalability. ● Generate a tabular comparison and line plot of centrality distributions. **Key Questions / Analysis / Interpretation to be evaluated during/after Implementation** **Que / Key Point**	CO2/PO1

	- How do different centrality measures behave across scale-free and random networks? - What is the time complexity of each centrality algorithm on large networks? - Which tool performs best in terms of efficiency and accuracy for different centralities? **Supplementary Problems (For Fast Learners)** - Extend analysis to weighted or directed graphs. - Implement parallel computation of centralities using external tools or libraries. **Key Skills to be addressed** - Graph generation and manipulation - Centrality analysis - Performance evaluation of algorithms - Data visualization **Applications** - Identifying influential nodes in social or information networks - Network resilience and vulnerability analysis - Traffic or resource flow optimization in infrastructure networks **Learning Outcome** Students will learn how to apply and interpret centrality metrics, compare algorithms, and assess scalability in large networks. **Dataset/Test Data (Source and Description If Applicable)** Synthetic scale-free and random networks generated using NetworkX, SNAP, or igraph. **Tools/Technology To Be Used** Python, NetworkX, SNAP, igraph	

	Total Hours of Problem Definition Implementation 4 hours **Total Hours of Engagement =** Implementation + modification + testing: approx. 6-8 hours **Post Laboratory Work Description** ● Submit tabular comparison of centrality scores for each type of network. ● Include time performance chart and plotted distributions. ● Submit Python code and graphs used for analysis. **Evaluation Strategy Including Viva** ● Viva on centrality concepts and application ● Assessment of code and result interpretations ● Clarity and depth in tabular and graphical comparison	
4	**Problem Definition:** Implement SIR and SIS diffusion models on real-world graphs. Simulate spread and analyze outcomes. **Tasks:** Run diffusion models, visualize spread progression, analyze role of high-degree nodes. **Key Questions / Analysis / Interpretation to be evaluated during/after Implementation:** How does network structure influence spread? Role of key nodes? **Supplementary Problems (For Fast Learners):** Add animation or try IC/LT models. **Key Skills to be addressed:** Simulation modelling, epidemic theory, plotting. **Applications:** Viral marketing, epidemic modelling.	CO3/PO3

	Learning Outcome: Simulate and analyze diffusion models in real networks. **Dataset/Test Data (Source and Description If Applicable):** Facebook Network **Tools/Technology To Be Used:** Python, NetworkX, matplotlib **Total Hours of Problem Definition Implementation:** 4 hours **Total Hours of Engagement = Implementation + modification + testing:** approx. 4 – 5 hours **Post Laboratory Work Description:** Include simulation plots, scripts, and analysis. **Evaluation Strategy Including Viva:** Viva on model logic and correct simulations.	
5	**Problem Definition:** Create bipartite graphs from affiliation data (e.g., users–movies). Project onto one-mode graphs and analyze. **Tasks:** Build bipartite graph, project, calculate metrics. **Key Questions / Analysis / Interpretation to be evaluated during/after Implementation:** What insights does projection reveal? Differences in network stats? **Supplementary Problems (For Fast Learners):** Analyze node similarity, clustering patterns. **Key Skills to be addressed:** Bipartite modeling, projection, degree analysis. **Applications:** Recommendation systems, affiliation networks.	CO3/PO2

	Learning Outcome: Analyze two-mode network structure and transform into one-mode analysis. **Dataset/Test Data (Source and Description If Applicable):** DBLP, MovieLens **Tools/Technology To Be Used:** Python, NetworkX **Total Hours of Problem Definition Implementation:** 3–4 hours Total Hours of Engagement = Implementation + modification + testing: approx. 4–5 hours **Post Laboratory Work Description:** Submit graphs and comparison of original vs projected metrics. **Evaluation Strategy Including Viva:** Viva on bipartite modelling and projection results.	
6	**Problem Definition:** Generate scale-free and random graphs. Apply centrality algorithms: Degree, Betweenness, Closeness, Eigenvector, PageRank. Compare performance across graph types. **Tasks:** ● Generate graphs, apply centrality metrics, compare results, visualize using plots. **Key Questions / Analysis / Interpretation to be evaluated during/after Implementation:** ● How do different centrality metrics highlight influence? Which metric is scalable? How does network type affect results? **Supplementary Problems (For Fast Learners):** ● Apply on real datasets; use weighted/directed versions.	CO3/PO2

	Key Skills to be addressed: ● Centrality computation, graph comparison, visualization. **Applications:** ● Influence detection, infrastructure optimization. **Learning Outcome:** ● Compare centrality algorithms and analyze their use in real-world settings. **Dataset/Test Data (Source and Description If Applicable):** ● Synthetic graphs using NetworkX, SNAP **Tools/Technology To Be Used:** ● Python, NetworkX, igraph, SNAP **Total Hours of Problem Definition Implementation:** 4 hours **Total Hours of Engagement = Implementation + modification + testing:** approx. 6–8 hours **Post Laboratory Work Description:** Submit comparative tables, plots, and Python code. **Evaluation Strategy Including Viva:** Viva on centrality concepts, correctness of plots and interpretations.	
7	**Problem Definition:** Detect network motifs like triangles and stars, analyze frequency.	CO3/PO2

	Tasks: Find motifs, compare with random graph. **Key Questions / Analysis / Interpretation to be evaluated during/after Implementation:** What motifs are common? What structure do they reveal? **Supplementary Problems (For Fast Learners):** Compare motif patterns across domains. **Key Skills to be addressed:** Motif recognition, subgraph matching. **Applications:** Behavior modeling, anomaly detection. **Learning Outcome:** Learn micro-level structure analysis of graphs. **Dataset/Test Data (Source and Description If Applicable):** Facebook dataset **Tools/Technology To Be Used:** Python, NetworkX **Total Hours of Problem Definition Implementation:** 4 hours **Total Hours of Engagement = Implementation + modification + testing:** approx. 6–7 hours **Post Laboratory Work Description:** Submit motif frequency charts and interpretation report. **Evaluation Strategy Including Viva:** Viva on motif concepts and outcomes.	
8	**Problem Definition:** Load large networks in Gephi. Apply layouts and compare effectiveness visually. **Tasks:** Try multiple layouts, highlight communities, use filters. **Key Questions / Analysis / Interpretation to be evaluated during/after Implementation:**	CO4/PO2

	Which layout reveals structure best? How does layout change insight? **Supplementary Problems (For Fast Learners):** Customize appearance or export for reporting. **Key Skills to be addressed:** Layout application, styling, visualization. **Applications:** Network dashboards, education visuals. **Learning Outcome:** Practice layout-based network analysis for communication. **Dataset/Test Data (Source and Description If Applicable):** SNAP - Twitter, Facebook **Tools/Technology To Be Used:** Gephi **Total Hours of Problem Definition Implementation:** 3–4 hours **Total Hours of Engagement = Implementation + modification + testing:** approx. 6–7 hours **Post Laboratory Work Description:** Submit visual graphs, .gephi file, and screenshots. **Evaluation Strategy Including Viva:** Evaluation of layout usage, clarity of visualization.	